

**Remarks/Arguments**

The Examiner is thanked for the courteous telephone interview granted Applicant's representative on October 6, 2004. This Response has been prepared pursuant to comments and suggestions made by the Examiner during the interview.

Claims 1, 3, 5-16, 18 and 20-43 are pending in the present application. Claims 2, 4, 17 and 19 have been canceled, and claims 1, 3, 5, 6, 16, 18, 20, 21, 31, 32, 33, 34 and 35 have been amended. No claims have been added. Applicant believes the claims currently in the case patentably distinguish over the cited art and are allowable in their present form, and respectfully request reconsideration in view of the above amendments and the following comments.

**I. Objection to the Abstract**

The Examiner has objected to the Abstract because it exceeds 150 words in length, and has reminded Applicant of the proper language and format for an Abstract.

By the present Amendment, the Abstract paragraph has been rewritten to be proper in language and format, and to be less than 150 words in length.

Therefore, the objection of the Abstract has been overcome.

**II. 35 U.S.C. § 102, Anticipation**

The Examiner has rejected claims 1-4, 7-15, 16-19, 22-30 and 31 under 35 U.S.C. § 102(e) as being anticipated by U.S. Publication US 2002/0138821A1 (Furman et al.). This rejection is respectfully traversed.

Claim 1 has been amended to incorporate subject matter recited in original claims 2 and 4, and to further clarify the invention. In rejecting the claims, the Examiner states, with respect to claims 1, 2 and 4:

**Per claim 1:**

Furman teaches:

- a method of porting a program from a first platform to a second platform ("for providing seamless porting of source code originally written...for use under the Microsoft Windows operating system...to

computers operating under other Operating Systems..." in the Abstract)

- converting at least one of filenames and a directory structure of the program from a first platform standard for the first platform to a second platform standard for the second platform (Note paragraph 0045 of the specification)
- storing the program for use with the second platform ("the files are sent to the second computer..." in paragraph 0011)

substantially as claimed.

Office Action dated July 15, 2004, page 3.

**Per claim 2:**

The rejection of claim 1 is incorporated, and further, Furman discloses a first platform standard including a flexible filename standard and the second platform standard including a restricted filename standard as claimed ("unlike the WOS, UOS, is a case sensitive OS...", in paragraph 0045. The WOS (Windows operating system) is not case sensitive; therefore it is a more flexible standard as opposed to the UOS (Unix operating system).)

Office Action dated July 15, 2004, page 3.

**Per claim 4:**

The rejection of claim 1 is incorporated, and further, Furman discloses a first platform standard including a flexible directory structure and a second platform including a restricted directory structure as claimed ("It should be noted that there are no disk drive names under UOS, at least not in the manner used under WOS..." in paragraph 0026)

Office Action dated July 15, 2004, page 4.

Claim 1, as amended herein, reads as follows:

1. A method of porting a program from a first platform to a second platform, comprising:

converting at least one of filenames and a directory structure of the program from a first platform standard for the first platform to a second platform standard for the second platform, wherein the first platform standard includes a first filename structure and a first directory hierarchy structure, wherein the second platform standard includes a second filename structure and a second directory hierarchy structure, and wherein at least one of the second filename structure is more restricted in length than the first filename structure and the second directory hierarchy structure is more restricted in hierarchy than the first

directory hierarchy structure; and  
storing the program for use with the second platform.

Furman does not disclose "converting at least one of filenames and a directory structure of the program from a first platform standard for the first platform to a second platform standard for the second platform, wherein the first platform standard includes a first filename structure and a first directory hierarchy structure, wherein the second platform standard includes a second filename structure and a second directory hierarchy structure, and wherein at least one of the second filename structure is more restricted in length than the first filename structure and the second directory hierarchy structure is more restricted in hierarchy than the first directory hierarchy structure", and, accordingly, does not anticipate claim 1.

Furman discloses a method and apparatus for porting source code originally written for use under the Microsoft Windows operating system (WOS) to computers operating under other Operating Systems such as the Unix operating system (UOS). The present invention, on the other hand, is directed to converting at least one of a first filename structure and a first directory hierarchy structure from a first platform to a second platform, "wherein at least one of the second filename structure is more restricted in length than the first filename structure and the second directory hierarchy structure is more restricted in hierarchy than the first directory hierarchy structure". In the present invention, for example, a program is ported from a Unix platform to an OS/400 platform.

In the present invention, porting is from a platform having less restrictive standards to a platform having more restrictive standards, whereas in Furman, porting is from a platform having more restrictive standards to a platform having less restrictive standards. As pointed out on page 4, lines 15-25 of the present application:

In this way, an application developer can develop application files without being limited to the most restrictive filesystem conventions. That is, a developer may make use of the more flexible filesystem conventions of Unix, for example, when developing application files and use the apparatus and method of the present invention to automatically handle converting these more flexible filenames and directory structures to the more restrictive filenames and directory structures of platforms to which the application is to be ported.

Initially, Furman does not disclose “wherein at least one of the second filename structure is more restricted in length than the first filename structure” as now recited in claim 1. In rejecting claim 3, The Examiner points to paragraph 0045 of Furman as disclosing shortening filenames in view of the statement therein that “the disk name (“c”) is stripped off of the full WOS path...”. However, as described in paragraph 0042-0044 of Furman:

```
#include c:\MyFiles\name.app
```

The path format used above is not a format recognized by the UOS. In the configuration menu, the WOS reference to a PC hard disk is replaced by a path format recognized by the UOS, for example:

```
“/usr/public”
```

As a result, the translated instructions results in the following format described in paragraph 0046;

```
#include/usr/public/myfiles/name.app.
```

Thus, although the disk name (“c”) is removed in Furman, “usr/public” is added resulting in an increased length for the second filename structure. Accordingly, Furman does not disclose that the second filename structure is “is more restricted in length than the first filename structure” as recited in claim 1.

Furman also does not disclose wherein “the second directory hierarchy structure is more restricted in hierarchy than the first directory hierarchy structure” as now recited in claim 1. The Examiner points to paragraph 0026 in Furman as disclosing “It should be noted that there are no disk drive names under UOS, at least in the manner used under WOS...”, and concludes that this statement evidences that the directory structure in the second platform is more restricted than the directory structure in the first platform in Furman. However, Furman does not disclose that the directory structure in the second platform is more restricted in hierarchy than the directory structure in the first platform as

recited in amended claim 1. Furman does not appear to discuss hierarchy at all.

Accordingly, claim 1 is not anticipated by Furman for this reason as well.

For all the above reasons, Furman does not anticipate claim 1, and withdrawal of the rejection thereunder is respectfully requested.

Claims 3 and 7-15 depend from and further restrict claim 1, and are also not anticipated by Furman, at least by virtue of their dependency.

Independent claims 16 and 31 have been amended in a manner similar to claim 1 and are also not anticipated by Furman for substantially the same reasons as discussed above with respect to claim 1. Claims 18 and 22-30 depend from and further restrict claim 16 and are also not anticipated by Furman, at least by virtue of their dependency.

Therefore, the rejection of claims 1-4, 7-15, 16-19, 22-30 and 31 under 35 U.S.C. § 102 has been overcome.

Furthermore, Furman does not teach, suggest, or give any incentive to make the needed changes to reach the presently claimed invention. Furman actually teaches away from the presently claimed invention because it does not teach or suggest "converting at least one of filenames and a directory structure of the program from a first platform standard for the first platform to a second platform standard for the second platform, wherein the first platform standard includes a first filename structure and a first directory hierarchy structure, wherein the second platform standard includes a second filename structure and a second directory hierarchy structure, and wherein at least one of the second filename structure is more restricted in length than the first filename structure and the second directory hierarchy structure is more restricted in hierarchy than the first directory hierarchy structure" as in the presently claimed invention. Absent the Examiner pointing out some teaching or incentive to implement Furman to achieve the present invention, one of ordinary skill in the art would not be led to modify Furman to reach the present invention when the reference is examined as a whole. Absent some teaching, suggestion, or incentive to modify Furman in this manner, the presently claimed invention can be reached only through an improper use of hindsight using the Applicant's disclosure as a template to make the necessary changes to reach the claimed invention.

### III. 35 U.S.C. § 103, Obviousness

The Examiner has rejected claims 5, 6, 20, 21 and 32-43 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Publication US 2002/0138821A1 (Furman et al.) in view of Applicant's admittance of prior art (APA). This rejection is respectfully traversed.

In rejecting the claims, the Examiner acknowledges that Furman does not explicitly disclose that the restricted directory structure is a nonhierarchical directory structure. The Examiner states, however, that Furman is not limited to the specific embodiment outlined in the application wherein the first platform is a WOS and the second platform is a UOS. In addition, the Examiner states:

Furthermore, as discussed by APA, the existence and use of the OS/400 operating system, which consists of a nonhierarchical filesystem, was well known to one of ordinary skill in the art at the time the invention was made. Consequently, it would have been obvious to one of ordinary skill in the art at the time the invention was made to port a program from a first hierarchical directory structure to a second nonhierarchical directory structure, such as OS/400, as Furman discloses that the porting ability of his invention may be applied to other OS's (paragraph 0005), thereby enabling a program written to be executed under one operating system to be executed under another OS.

Office Action dated July 15, 2004, pages 7 and 8.

Independent claim 32 as amended herein is as follows:

32. A method of porting a program from a first platform to a second platform, comprising:

converting filenames and a directory structure of the program from a first platform standard for the first platform to a second platform standard for the second platform, wherein the first platform standard includes a hierarchical directory structure and the second platform standard includes a nonhierarchical directory structure; and

storing the program for use with the second platform, wherein the method is performed in a build environment.

Even if, as indicated by the Examiner, Furman may state that his invention is applicable to other OS's, Furman only discloses porting to a platform having a hierarchical directory structure. The reference does not disclose or suggest porting from a

first platform to a second platform "wherein the first platform standard includes a hierarchical directory structure and the second platform standard includes a nonhierarchical directory structure" as recited in claim 32; and does not appreciate advantages of doing so as described above.

Furthermore, although the OS/400 operating system may have been known at the time the present invention was made, as asserted by the Examiner, simply knowing of the OS/400 operating system and the Furman reference would not teach one skilled in the art how to make a conversion from a first platform standard to a second platform standard "wherein the first platform standard includes a hierarchical directory structure and the second platform standard includes a nonhierarchical directory structure" as recited in claim 32. In fact, as indicated previously, Furman only teaches the opposite process, i.e., Furman teaches porting from a platform having more restrictive standards to a platform having less restrictive standards; and, accordingly, actually teaches away from combining Furman and APA as proposed by the Examiner. Accordingly, one of ordinary skill in the art would not be inclined to use Furman as a starting point to solve the problem solved by the present invention.

For at least all the above reasons, it would not be obvious to combine Furman and APA to achieve the present invention. Claim 32, accordingly, is not obvious over Furman in view of APA, and should be allowable in its present form.

Claims 33-43 depend from and further restrict claim 32, and are also not unpatentable over Furman in view of APA, at least by virtue of their dependency. Claims 5, 6, 20 and 21 depend from and further restrict either independent claim 1 or independent claim 16, and are also not unpatentable over Furman in view of APA, at least by virtue of their dependency.

Therefore, the rejection of claims 5, 6, 20, 21 and 32-43 under 35 U.S.C. § 103 has been overcome.

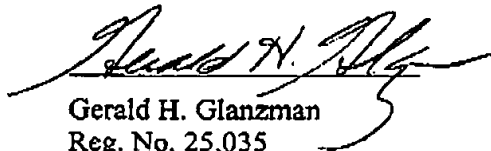
**IV. Conclusion**

It is respectfully urged that claims 1, 3, 5-16, 18 and 20-43 patentably distinguish over the cited art and are allowable in their present form. This application is, accordingly, believed to be in condition for allowance, and it is respectfully requested that the Examiner so find and issue a Notice of Allowance in due course.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: November 12, 2004

Respectfully submitted,



Gerald H. Glanzman  
Reg. No. 25,035  
Yee & Associates, P.C.  
P.O. Box 802333  
Dallas, TX 75380  
(972) 385-8777  
Attorney for Applicant